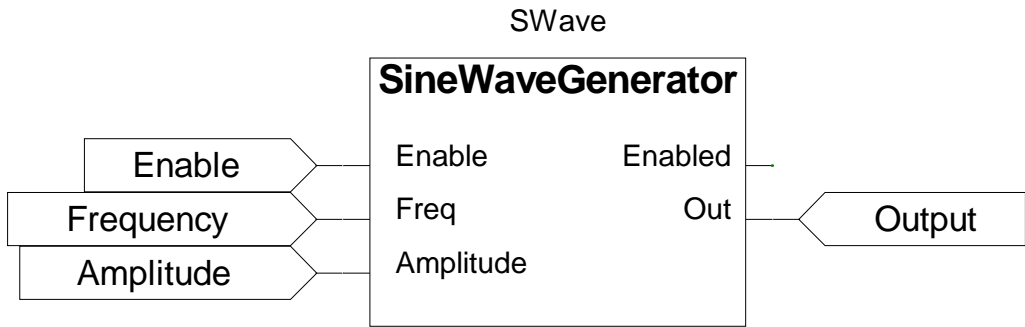


```

VAR
SWave : SineWaveGenerator;
Enable : BOOL;
Frequency : REAL := 0.1;
Amplitude : REAL := 1;
Output : REAL;
END_VAR
    
```

1



	Project : SineWave	
	PROGRAM : SineWave	
	Release : SineWave	Ver :1.00
	Author :	Date:19/03/2013
	Note :	Page:1 of 1

```

VAR_INPUT
Enable : BOOL; (* Abilita l'esecuzione *)
Freq : REAL; (* Frequenza dell'onda [Hz] *)
Amplitude : REAL; (* Ampiezza dell'onda *)
END_VAR

VAR_OUTPUT
Enabled : BOOL; (* FB Enabled *)
Out : REAL; (* Output value *)
END_VAR

VAR
TimeBf : ARRAY[ 0..1 ] OF UDINT; (* Time buffer (uS) *)
Period : REAL; (* Wave period (uS) *)
Angle : REAL; (* Omega vector angle (Rad) *)
TimePsd : UDINT; (* Time passed (uS) *)
END_VAR
    
```

```

1 (* ***** *)
2 (* FUNCTION BLOCK "SineWaveGenerator" *)
3 (* ***** *)
4 (* Questo blocco funzione genera un'onda sinusoidale. *)
5 (* ----- *)
6
7 (* ----- *)
8 (* GESTIONE ABILITAZIONE *)
9 (* ----- *)
10 (* Gestione disabilitazione. *)
11
12 IF NOT(Enable)THEN
13
14     (* One shot disabilitazione FB. *)
15
16     IF (Enabled) THEN
17         Enabled:=FALSE; (* FB Enabled *)
18         Out:=0.0; (* Output value *)
19     END_IF;
20
21     RETURN;
22 END_IF;
23
24 (* Gestione one shot abilitazione. *)
25
26 IF NOT(Enabled)THEN
27     Enabled:=TRUE; (* FB Enabled *)
28     TimeBf[1]:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
29     Period:=(1.0/Freq)*1000000.0; (* Wave period (uS) *)
30 END_IF;
31
32 (* ----- *)
33 (* GESTIONE ABILITAZIONE *)
34 (* ----- *)
35 (* Calcolo tempo trascorso da ultima esecuzione. *)
36
37 TimeBf[0]:=SysGetSysTime(TRUE); (* Time buffer (uS) *)
38 TimePsd:=TimePsd+(TimeBf[0]-TimeBf[1]); (* Time passed (uS) *)
39 TimeBf[1]:=TimeBf[0]; (* Time buffer (uS) *)
    
```

Project : SineWave	
FUNCTION BLOCK : SineWaveGenerator	
Release : SineWawe	Ver :1.00
Author :	Date:19/03/2013
Note :	Page:1 of 2

FUNCTION_BLOCK SineWaveGenerator

```

40
41 (* Riconduco il tempo al periodo definito dell'onda. *)
42
43 IF (TimePsd >= TO_UDINT(Period)) THEN
44     TimePsd:=TimePsd-TO_UDINT(Period); (* Time passed (uS) *)
45 END_IF;
46
47 (* ----- *)
48 (* Calcolo angolo in radianti relativo al tempo trascorso. Se l'onda ha *)
49 (* frequenza pari a Freq in hertz, significa che la velocità angolare del *)
50 (* vettore (omega) è 2*pi radianti per ogni periodo T (cioè 1/Freq). *)
51 (* Quindi la proporzione è: 2*pi:T = X:TimePsd *)
52 (* Dove:2*pi=2 pigreca radianti cioè 360 gradi *)
53 (*     T=periodo cioè 1/Freq *)
54 (*     TimePsd=Tempo trascorso in microsecondi *)
55 (* ----- *)
56 (* Eseguo calcolo dell'angolo. *)
57
58 Angle:=(6.28318*TO_REAL(TimePsd))/Period; (* Omega vector angle (Rad) *)
59
60 (* Calcolo il seno dell'angolo e lo moltiplico per l'ampiezza. *)
61
62 Out:=SIN(Angle)*Amplitude;
63
64 (* [End of file] *)
65
66

```

	Project : SineWave	
	FUNCTION BLOCK : SineWaveGenerator	
	Release : SineWawe	Ver :1.00
	Author :	Date:19/03/2013
	Note :	Page:2 of 2